Rapport projet synthétiseur

Table des matières

[Introduction 2](#_Toc103001478)

[Scénario d’usage du synthétiseur 2](#_Toc103001479)

[Bilan d’entrée/sortie 3](#_Toc103001480)

[Bilan des périphériques 4](#_Toc103001481)

[Bilan de la mémoire 4](#_Toc103001482)

[Liste des tâches de l’OS avec priorité 5](#_Toc103001483)

[Schémas bloc des périphériques 6](#_Toc103001484)

[Schémas électriques 7](#_Toc103001485)

Jérémy COLZY

Mathieu GRESSET

Mamadou BARRY BOY

Léo DELAS

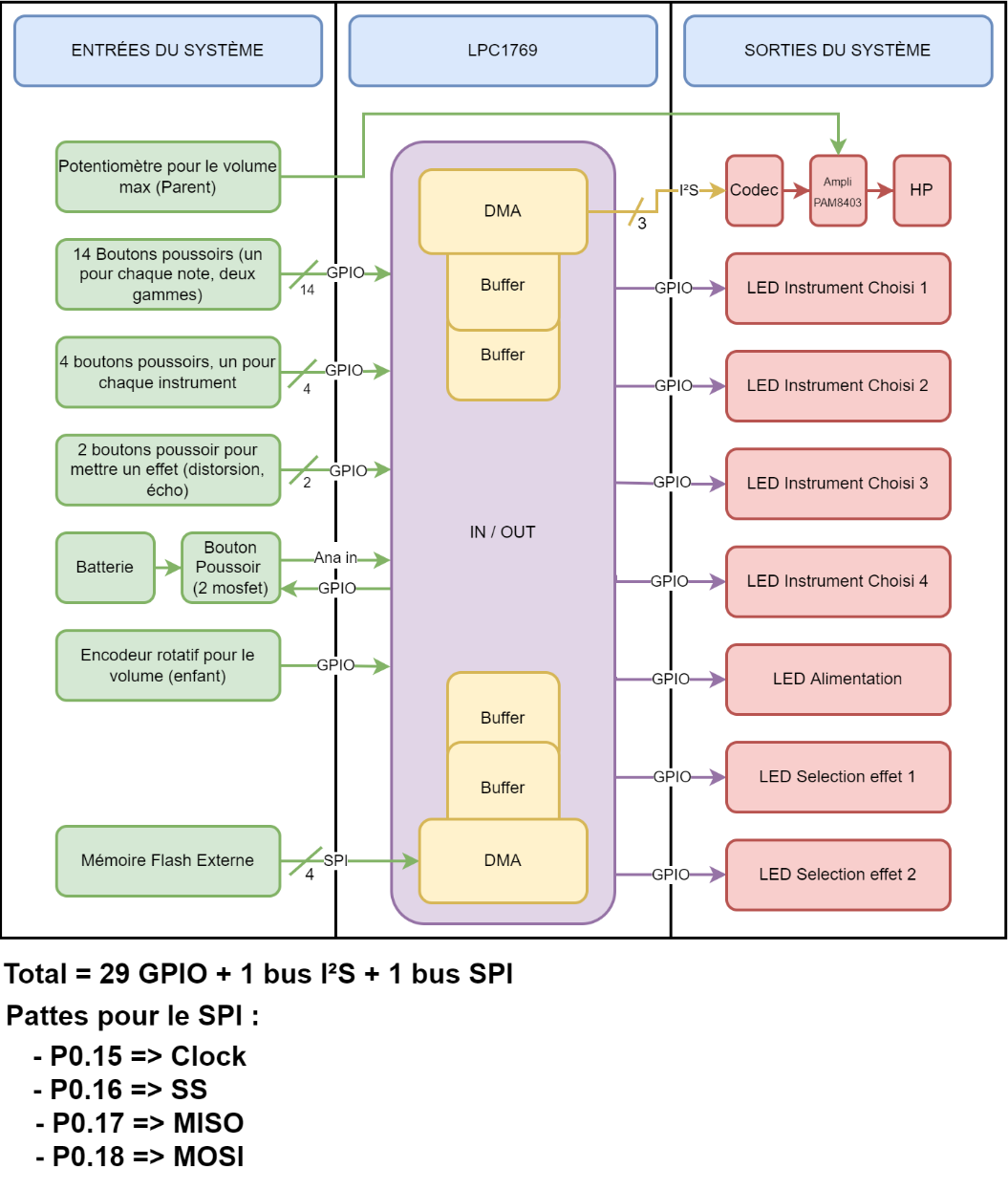
## Introduction

Le but de ce document est de détailler les solutions choisies concernant la configuration du LPC1769 ainsi que les périphériques extérieurs pour la réalisation du synthétiseur, à savoir le scénario d’usage, le bilan d’entrée/sortie, le bilan de la mémoire, la liste des tâches de l'OS avec priorité, schéma bloc des périphériques, les schémas électriques, une liste des interruptions, et une liste d’avancement du projet.

## Scénario d’usage du synthétiseur

* L’adulte ouvre le boîtier d’alimentation afin d’accéder au bouton permettant de régler le volume maximum de l’appareil puis referme le boîtier d’alimentation.
* L’enfant voit le synthétiseur, il essaie de jouer un son en appuyant sur une touche du synthétiseur, mais cela ne fonctionne pas, il appui donc sur le bouton ON/OFF.
* Au bout de quelques millisecondes, le produit est allumé et fonctionnel, prêt à être utilisé
* L’enfant aperçoit 4 LEDs avec une image d’instrument et un bouton associer à chaque LEDs, par défaut la LED du piano est allumée, en appuyant sur un bouton pour changer l’instrument, la LED correspondante s’allume et celle déjà allumée s’éteint.
* L’enfant appuie donc sur un bouton pour jouer un son, une fois appuyer le son correspondant est joué.
* Dès que la touche est relâchée, le son disparait.
* Lorsque l’enfant appuie sur plusieurs touches à la fois, les sons s’additionnes. Au maximum, 4 sons peuvent être joué simultanément. Si une touche supplémentaire est appuyée, le son de la première touche appuyé disparait.
* Il peut aussi ajouter des effets de sons tel qu’un écho ou de la distorsion.
* Le volume maximum du synthétiseur peut être défini par un adulte à l’aide d’un potentiomètre protégé par un cache pour pas que les enfants y touche.
* Les enfants ont la possibilité de régler le volume du synthétiseur via un encodeur rotatif. Le volume varie entre 0 dB et le volume défini par les parents par pas de 3 dB.

## Bilan d’entrée/sortie

Voici ci-dessous un schéma de notre bilan d’entrée/sortie pour le projet du synthétiseur pour enfant.

Notre système comporte 23 entrées dont une non reliée au microcontrôleur (Potentiomètre pour le volume max) qui est directement relié à l’amplificateur pour régler le gain de l’amplificateur. Le système comporte également 7 LED de sortie afin de communiquer des informations sur l’état du système.

Les données de la mémoire flash sont chargé dans des buffers de la mémoire interne de la carte LPC1769 à l’aide d’un DMA qui est un périphérique interne à la carte.

Les échantillons audio traités sont stocké dans les buffers de la carte LPC1769 qui sont ensuite envoyé au codec audio.

Une fois le signal reçu sur le codec, le signal est traduit et envoyé à l’amplificateur qui adapte le niveau de sons avant que le haut-parleur ne joue le son.

## Bilan des périphériques

Les périphériques que nous allons utiliser pour ce projet sont les suivants :

* 2 DMA
  + 1 pour récupérer les données de la mémoire flash sur la carte LPC1769
  + 1 pour envoyer les données de la carte LPC1769 au circuit qui permet de jouer le son
* 1 Timer
  + 1 Timer pour gérer le système de veille automatique.

## Bilan de la mémoire

Après avoir testé plusieurs méthodes d’échantillonnage/stockages nous avons décidé de stocker les sons de tous les instruments pour chaque note en brute dans la mémoire. Le seul traitement du signal qui sera à opérer concernera les effets appliqués aux notes.

Monsieur Bras nous a fourni une mémoire de 16 Mo. Suite à nos essais nous avons déterminé que pour stocker toutes les 14 notes (soit 2 octaves) pour chacun des 4 instruments il nous faut 11 Mo d’espace mémoire.

Nous avons, sur la carte, une SRAM de 16 ou 32 Ko. Le but pour les buffers est de ne pas dépasser la moitié de la capacité de la SRAM.

Moitié = 8 Ko,

4 buffers = 2 Ko par buffers en 16 bits on a donc 1024 échantillons de buffer

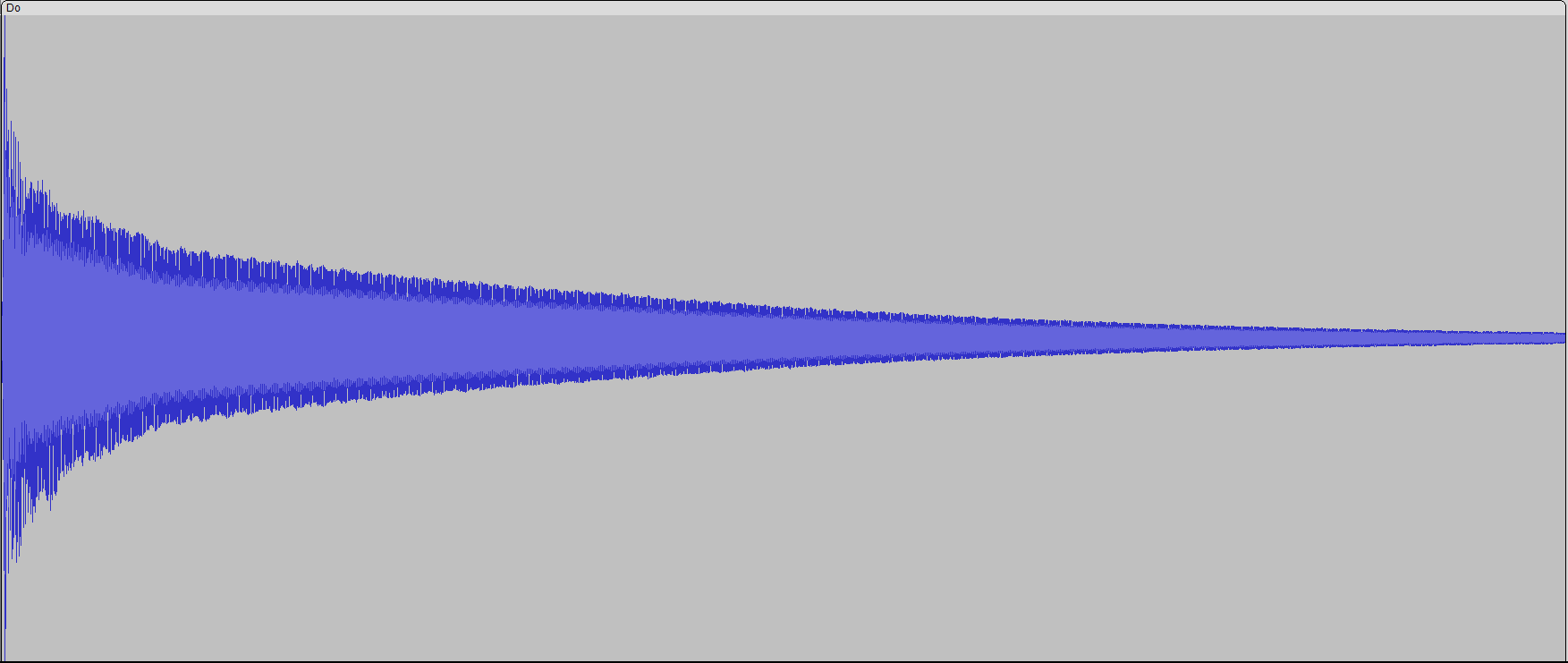
## Génération des notes de musiques

Afin de générer les sons depuis le synthétiseur, nous avons choisi d’enregistrer les différentes notes des 4 instruments choisis afin de les stockés dans la mémoire. Pour rappel, les instruments choisis sont :

* Le piano
* La flute de pan
* Le carillon
* Le saxophone

Ces différents instruments sont classé en deux catégories, les instruments évanescents (piano et carillon) et les instruments non évanescent (flute et saxophone). Pour chacun, nous avons enregistrer les notes des octaves 3 et 4. Nous n’avons pas voulu enregistrer les notes modifiées par des altérations (dièse ou bémol). Cela représente 14 notes en tout.

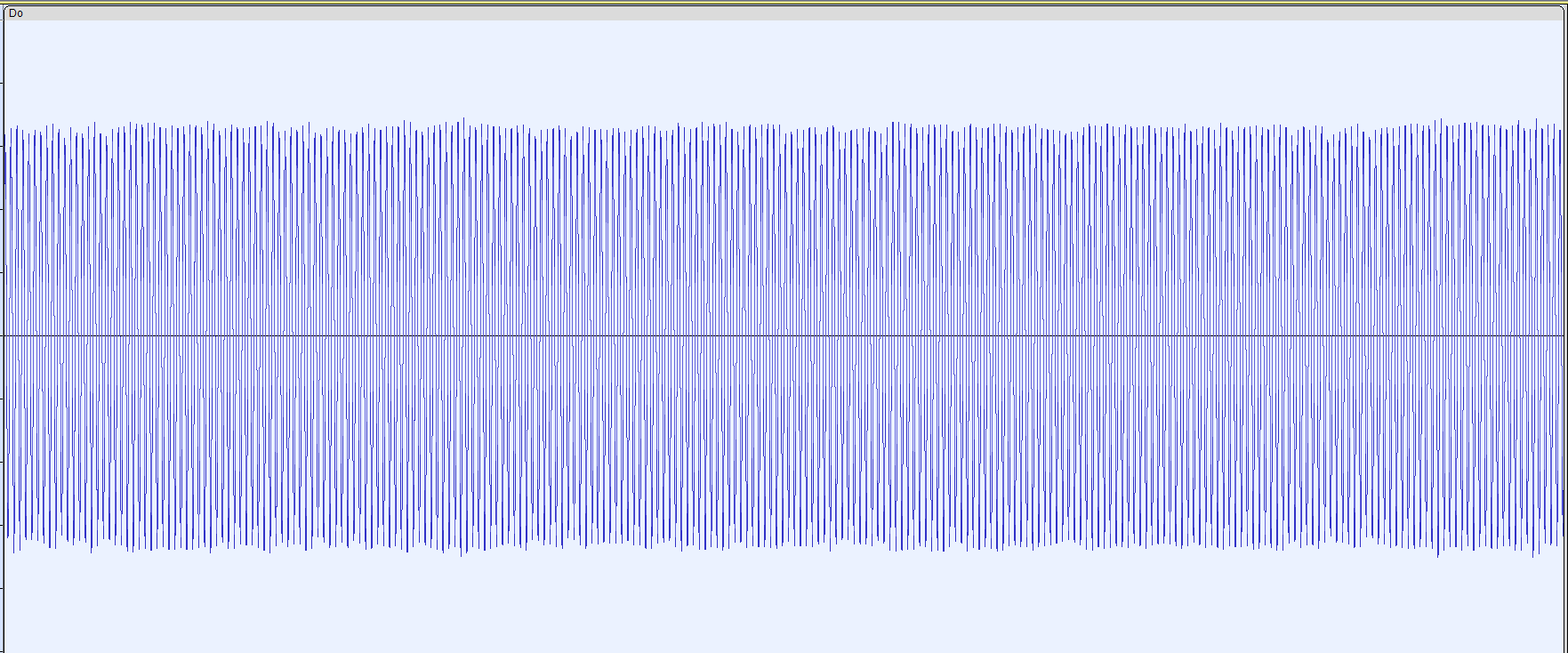
Pour les instruments évanescent, nous avons choisis d’enregistrer le son de chaque note entre le temps d’appuie sur une touche et le temps où le son disparait.



Chronogramme de la note Do (octave 3) du piano.

Un tel son sera alors stocké ainsi dans la mémoire.

Pour les instruments non évanescent, nous avons enregistré deux types de son, une note pendant une durée de 1 secondes environs et une seule période de son. Ces deux enregistrement peuvent être utile selon la taille de la mémoire. Si la mémoire est trop petite, nous stockerons seulement une période de son qui sera répété tant que la touche est maintenue appuyée. Cela demande une grande quantité de ressource au processeur. Si la mémoire est suffisamment grande, nous stockerons le son d’une durée de 1 seconde et celui-ci sera lu en boucle tant que la touche est appuyée. Ce système permet d’économiser des ressources processeur mais nécessite plus de place mémoire.



Chronogramme d’une note Do (octave 3) de flute de pan sur une durée de 1s.

Une image contenant musique

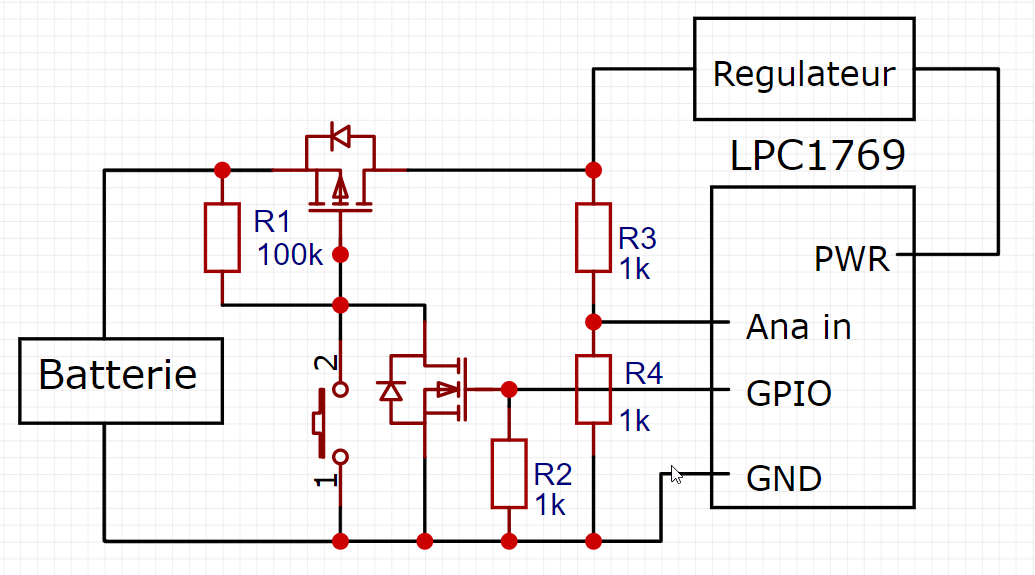
Description générée automatiquement

Chronogramme d’une période de note Do (octave 3) de flute de pan.

## Liste des tâches de l’OS avec priorité

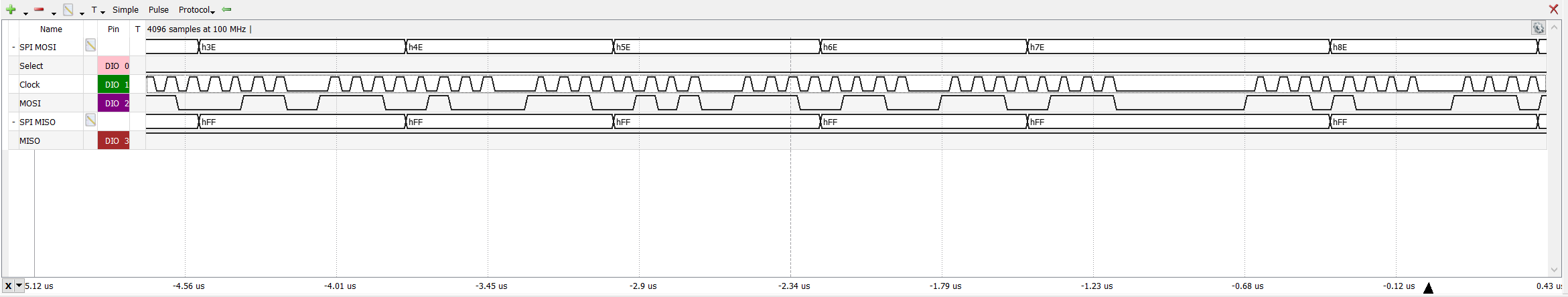
## Schémas bloc des périphériques

## Schémas électriques

Pour la gestion de l’alimentation nous utiliserons le schéma électrique suivant.

Le but de ce schéma électrique est d’alimenter la carte du LPC1769 lors d’un appuie sur le bouton d’alimentation, puis d’avoir un maintien de l’alimentation sans avoir à rester appuyé sur le bouton. Lorsque la batterie descend en dessous d’un certain seuil, la carte coupe son alimentation en bloquant le transistor de maintien de l’alimentation. La carte LPC1769 implémentera également une veille automatique au bout d’un certain temps.

## Captures d’oscilloscopes (SPI)

Capture d’oscilloscope d’une transmission spi d’une même valeur

capture d’oscilloscope d’une transmission spi d’un buffer incrémentale

## Capture d’oscilloscope (Bouton Poussoir)

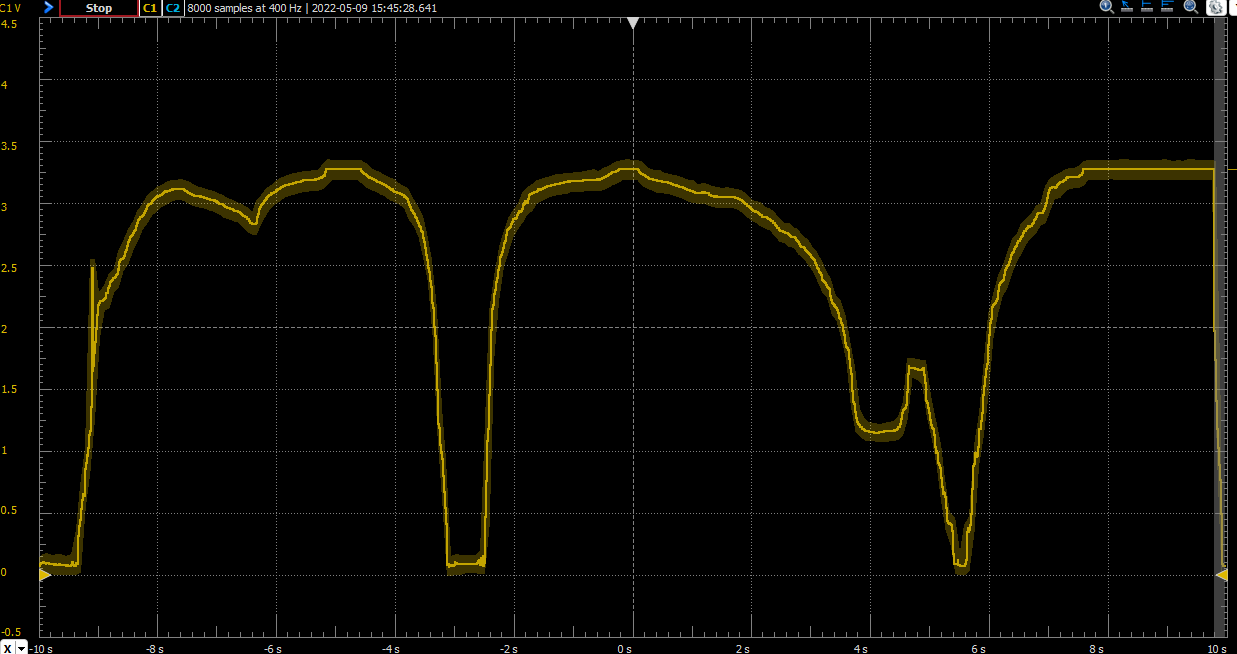
Appui du bouton poussoir. Allumage de la Led rouge de la carte.

## Capture d’oscilloscope (Encodeur incrémental)

Décrémentation du codeur par pas de 0x10

Incrémentation du codeur par pas de 0x10

## Capture d’oscilloscope (Potentiomètre)



Le potentiomètre utilisé n’est pas linéaire d’où la forme logarithmique de la courbe lors des phase de montée ou de descente.

Diminution du potentiomètre

Augmentation du potentiomètre